

Отчет по лабораторной работе №2

Выполнил: Вампилов Буда Арсаланович 411032, гр. М4130

BitsCount

В реализации примера решения закодированы в виде битовой строки определенной длины, объекта класса BitString.

Результаты расчёта количества итераций алгоритма от размерности проблемы при размере популяции 1000 (для меньшего размера функционал сходится очень долго).

Размерность	Run 1	Run 2	Run 3	Run 4	Run 5	Среднее
20	11	14	14	11	13	12,6
50	389	455	74	452	334	340,8
100	1973	2632	3228	2899	2835	2713,4

Travelling salesman problem

Найденный кратчайший путь относительно параметров алгоритма.




Population size	Elitism	Number of generations	Selection strategy	Route	Total Distance
300	3	100	Truncation Selection (50%)	Rome -> Athens -> Vienna -> Berlin -> Helsinki -> Stockholm -> Copenhagen -> Amsterdam -> Brussels -> Luxembourg -> Paris -> London -> Dublin -> Lisbon -> Madrid -> Rome	10494
200	3	100	Truncation Selection (50%)	Madrid -> Lisbon -> Dublin -> London -> Paris -> Luxembourg -> Brussels -> Amsterdam -> Copenhagen -> Stockholm -> Helsinki -> Berlin -> Vienna -> Athens -> Rome -> Madrid	10494
100	3	100	Truncation Selection (50%)	Lisbon -> Madrid -> Rome -> Athens -> Vienna -> Berlin -> Helsinki -> Stockholm -> Copenhagen -> Amsterdam -> Brussels -> Luxembourg -> Paris -> London -> Dublin -> Lisbon	10494
10	3	100	Truncation Selection (50%)	Madrid -> Rome -> Athens -> Helsinki -> Stockholm -> Copenhagen -> Amsterdam ->	11789

				Brussels -> Berlin -> Vienna -> Luxembourg -> Paris -> London -> Dublin -> Lisbon -> Madrid	
300	30	100	Truncation Selection (50%)	Copenhagen -> Amsterdam -> Brussels -> Luxembourg -> Paris -> London -> Dublin -> Lisbon -> Madrid -> Rome -> Athens -> Vienna -> Berlin -> Helsinki -> Stockholm -> Copenhagen	10494
300	3	1000	Truncation Selection (50%)	Athens -> Vienna -> Berlin -> Helsinki -> Stockholm -> Copenhagen -> Amsterdam -> Brussels -> Luxembourg -> Paris -> London -> Dublin -> Lisbon -> Madrid -> Rome -> Athens	10494
300	3	100	Tournament Selection ($p = 0.95$)	Amsterdam -> Copenhagen -> Stockholm -> Helsinki -> Berlin -> Vienna -> Athens -> Rome -> Madrid -> Lisbon -> Dublin -> London -> Paris -> Luxembourg -> Brussels -> Amsterdam	10494
300	3	100	Sigma Scaling	Lisbon -> Madrid -> Rome -> Athens -> Vienna -> Berlin -> Helsinki -> Stockholm -> Copenhagen -> Amsterdam -> Brussels -> Luxembourg -> Paris -> London -> Dublin -> Lisbon	10494
300	3	100	Roulette Wheel Selection	Madrid -> Rome -> Athens -> Vienna -> Berlin -> Helsinki -> Stockholm -> Copenhagen -> Amsterdam -> Dublin -> London -> Brussels -> Luxembourg -> Paris -> Lisbon -> Madrid	10858

Mona Lisa

Результаты оптимизации подбора полигонов для воспроизведения картины.

Решение	Итерация	Фитнесс	Кол-во полигонов и углов	Рисунок
---------	----------	---------	--------------------------	---------

плохое	5415	292550	23 и 120	 <p>23 polygons, 120 vertices</p>
среднее	12116	269810	17 и 126	 <p>17 polygons, 126 vertices</p>
хорошее	41314	223575	39 и 283	 <p>39 polygons, 283 vertices</p>

Ответы на вопросы

1. Типы задач
 - 1.1. BitsCount. Задача одной функции в дискретном (бинарном) пространстве.
 - 1.2. Traveling Salesman Problem. Задача многокритериальной оптимизации (нужно посетить каждый город и при этом найти кратчайший путь) в дискретном пространстве (если расстояния между городами заданы заранее).
 - 1.3. Mona Lisa. Задача мультимодальной минимизация одной функции (несколько комбинаций полигонов дают один результат, т. е. функция имеет несколько оптимумов) в непрерывном пространстве координат углов полигонов.

2. Списки строк кандидатов List<String> candidate и расстояние totalDistance полученное после применения фитнес-функции getFitness.
3. Генотип - список полигонов List<ColouredPolygon>, фенотип - отрендеренное растровое изображение.